Available online at www.eccomasproceedia.org
Eccomas Proceedia UNCECOMP (2023) 453-465

ECCOMAS
Proceedia

UNCECOMP 2023
5th ECCOMAS Thematic Conference on
Uncertainty Quantification in Computational Science and Engineering
M. Papadrakakis, V. Papadopoulos, G. Stefanou (eds.)
Athens, Greece, 12-14 June 2023

ENHANCING TRANSFER LEARNING FOR CRASHWORTHINESS STUDIES UNDER LOW DATA AVAILABILITY THROUGH SPHERE PROJECTION

Giada Colella[1,2], Volker A. Lange[1], and Fabian Duddeck[2]

[1]BMW Group, Research and Innovation Center
Knorrstraße 147, Munich, Germany
e-mail: giada.colella@bmw.de

[2] Technical University of Munich, School of Engineering and Design
Arcisstraße 21, Munich, Germany

Abstract. *This work proposes a transfer learning (TL) approach for crashworthiness analyses. It investigates the possibility to exploit data and geometrical information coming from past studies when only a few data points are available for the current product of interest. This situation occurs early on in the development, when assessing the crash safety in the automotive industry. As the final characteristics of the product under investigation are not yet entirely defined, TL can be of aid. By transferring the knowledge from past simulations or uncertainty propagation studies — the so-called source domain — to forthcoming cases — the target domain, TL can offer valuable predictions. The data coming from past similar products, however, often belong to different geometrical designs with respect to the one of interest. The TL network needs to be able to distinguish them. Therefore, the methodology can be enhanced by coupling the TL with an already existing geometry classification technique: sphere projection. In this way, the TL approach becomes able to predict the new structural behavior under uncertainties. We demonstrate this by applying TL to an explicatory bonnet example. The obtained results picture the enhanced TL approach as an attractive starting point. With respect to past studies, the proposed approach gives more flexibility to TL for crash analyses. It can now help to enhance UQ studies for crashworthiness when only a few data are available.*

Keywords: Transfer Learning, Crashworthiness, Low Data Availability, Sphere Projection.

ISSN:2623-3339 © 2023 The Authors. Published by Eccomas Proceedia.
Peer-review under responsibility of the organizing committee of UNCECOMP 2023.
doi: 10.7712/120223.10349.19593

1 INTRODUCTION

Assessing vehicle safety is a complicated, yet fundamental task. Early on in the product development, car manufacturers face the tough challenge of low data availability. Nevertheless, they need to ensure the compliance of the vehicles with strict safety requirements.

At the very beginning of a product development process, engineers draw upon their expertise: a new product is often considered as a broad combination — and modification — of existing ones. The ultimate geometrical and material data are not entirely defined, and the prototype of the vehicle is not yet available. Full vehicle hardware crash tests cannot be performed and, due to these circumstances, finite element (FE) simulations aid the design process.

Crash FE simulations have become essential for automotive R&D. Engineers simulate extensively in the early-stage of the vehicle development. In this way, they get the preliminary insights regarding the crashworthiness of the new design. Performing uncertainty quantification (UQ), investigating design reliability and robustness, as well as addressing design and concept feasibility, however, becomes impractical with respect to time, as multiple simulations need to be performed.

To overcome these difficulties, metamodels are often considered. These simplified approaches are known to relieve the computational effort of simulation-based problems, allowing a faster output generation [1]. Despite the provided advantages, however, the procedure that leads to the calculation of a metamodel is often hard. It requires a high number of computationally expensive simulations of the initial system: numerous simulation data need to be produced, stored and classified in order to correctly train and validate the metamodel. These data are not easy to be collected when, in the early-stage design, still little-to-no insights of the product are available.

These circumstances can benefit from transfer learning (TL), which can be seen as an attractive data-driven alternative to expert knowledge. The field of TL aims to leverage knowledge from a related source domain to improve the learning performance in a target domain [2]. This approach is suitable for the application at hand: the large availability of predecessor designs or UQ data in the source domain supports the calculation of a predictive model for the new product under development in the target domain. The favorable outcome of TL, however, depends on the definition of the two domains [3].

The methodology can lead to a positive or negative transfer depending on the similarity between the two domains. In the investigated case, source and target differ in terms of geometry. An immediate idea to avoid negative transfer is, therefore, to add geometrical information that help to differentiate between source and target. For TL to work, this geometrical features description has to be uniform for both the domains. The aim of the present study is to investigate how, and if, the geometrical information of source and target domain can be included in the TL framework in order to enhance the learning performance.

To answer to this research question, we propose a methodology that merges TL with the sphere projection (SP) method. This mathematical technique allows to efficiently represent geometrical data — in the form of point clouds — as matrices. Here, we embed SP in the TL framework to allow the geometrical representation of the source and target domains and, consequently, to improve the performance of the black-box approach.

As an example, the technique is applied to bonnet structures. The geometries of the previously developed bonnets, as well as the necessary pieces of information for the UQ, constitute the source domain; the few available data coming from the FE simulations of the interest product, instead, create the target domain. Merging TL and SP, we aim to propose a methodology to automatize the decision making process which, in the early phase of crashworthiness studies, engineers partially follow based on their expertise.

Providing a cost-effective predictive model for the new product, TL can become a powerful method for multipurpose metamodeling. Its usage can be exploited not only for UQ, but also for verification and validation (V&V), concept feasibility or robustness studies. Embedding TL with SP, this paper opens up the possibility to include, beyond just the geometry representation, the one of the overall FE simulation. Combining white-box — FE simulations — and black-box — TL — approaches, this paper is the preliminary step that anticipates a grey-box approach.

2 STATE OF THE ART

In crashworthiness studies, the use of TL appears worthy of pursuit. In contrast to well-established ML techniques, TL allows to rely on a limited amount of data in the domain of interest [4]. This is advantageous because, while the availability of FE simulations is already limited in the early phase of the vehicle development, experimental data are not available at all.

TL first learns the core relationships from a source domain, and then readapts to the few data of a target domain. If the two domains are too different from each other, however, the predictive performance on the target domain can be negatively affected by the procedure [4]. To avoid this phenomenon, known as negative transfer, we propose to enhance TL by means of the SP technique. SP is thought to serve as a method to uniformly describe the geometries representative of the two domains. To lay the groundwork for their connection, the basic theory of both TL and SP is summarized in the following.

2.1 Transfer learning

Transfer learning (TL), also referred to as "domain adaptation" [5], is the approach that helps machine learning (ML) techniques to relax the need of collecting big amounts of highly reliable data. TL allows a ML model — trained to solve one problem — to be adapted to solve another problem [3]: in Fig. 1, respectively, step I and step II. Under the assumption that the two domains are similar, TL relies on extracting the basic knowledge from the so-called source domain, and applying it to fulfil the required task in the target domain [4]. Depending on its application, TL can be implemented in different ways.

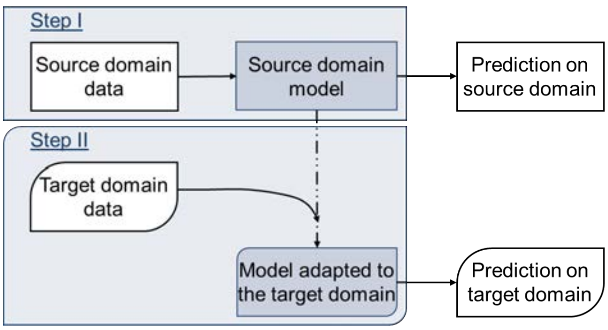

Figure 1: Transfer learning scheme.

Many literature sources focus on the application of TL to ML-typical problems, like computer vision [6], sentimental analysis [7], or visual categorization [8]. In these cases, TL is implemented modifying pre-trained deep learning models, e.g. Inception V3 or VGG. These layered architectures are known to learn different features at different levels. In [9], for example,

the authors show that the final layer of a deep learning system can work as a feature-extractor. This modifying of an existing model might prove inadequate for industrial use-cases due to the involved data types. A common procedure to perform domain adaptation in these cases is explained in [10]: first, a multilayer perceptron neural network (NN) is trained on the source domain; next, its architecture and hyperparameters are adapted to the target domain.
As of yet, industrial applications of TL appear to be sparsely reported in literature. Few studies, however, have already highlighted the strong potentialities of this technique in industrial scenarios, e.g. [5, 11, 12, 13]. Industry offers frequent situations characterized by low availability of data, providing a promising outline for the TL implementation.
As a representative example, the authors of [14] apply TL to structural mechanics, electrochemistry, and fluid mechanics. They propose three different ways to perform the domain adaptation: the bi-fidelity TL — BFTL-1 and 2, and the bi-fidelity weighted learning — BFWL. In BFTL-1, the uppermost hidden layer of a network trained using the source dataset is fine-tuned using the target dataset; in BFTL-2, a small network is added to model the map between the output estimated by the source network and its target counterpart; in BFWL, the parameters of the initial NN are modified according to data generated via a target trained Gaussian process. The study shows the efficiency of TL in diverse study cases when only small amounts of target samples — e.g. 20 instead of 200 — are given.
The benefit of TL depends on the quality of the pre-trained model, as well as the definition of the two domains. Moreover, attention should be paid to how transferable, and under which conditions, the features between the different domains are. This topic is discussed in great detail in [3].

2.2 Sphere projection

The literature offers a wide assortment of geometry classification methods, based on disparate techniques with specific characteristics, e.g. voxelization [15], multi-view [16], signed distance functions [17], etc. Methods based on voxelization, for instance, require high computational effort to process the data; multi-view based techniques are easily applicable to complex components, but mainly show their potential when the components are convex. Their appeal generally depends on the specific application or task at hand.

A previous attempt to include geometrical information in the TL workflow is proposed in [18], where geometrical features are used to take into account the geometrical differences between the domains. If the geometry of the studied components is sufficiently simple, one can describe their features directly: width, height, thickness, etc. This approach was successfully used in [18] for a crash box structure. It does, however, become infeasible when faced with higher complexity. The sphere projection (SP) technique is worth to be investigated as it provides an attractive compromise between computational costs and variety of geometries it can be applied to.

The SP method is firstly introduced in [19, 20] as a mathematical technique that allows to convert a 3D point cloud into a unique signature — a 2D matrix. The method consists of the following steps. First, forming a sphere — namely, the spherical detector surface — around the 3D geometry under investigation. The center of the detector surface coincides with the center of gravity of the geometry; one then divides the surface of the sphere into parts — referred to as pixels. The step at the heart of the method is to then project the nodes of the interest geometry onto the surface of the sphere. The count of projected nodes per pixel [21] is then stored as entries in a 2D matrix. This yields the unique signature of the original 3D geometry.

SP is often coupled to a widely used dimensionality reduction technique: the principal component analysis (PCA) [21]. With PCA, the principal axes of the considered geometries are computed and then aligned with the coordinate axes of the 3D space. In this way, PCA allows to have a consistent representation of different geometries regardless to their orientation in space. The nodes of the oriented geometry become the inputs for the SP method. PCA also allows the computational cost of the SP method to remain unchanged.

This conjunction of SP with PCA was employed in [21] to automatize part recognition. In [22], the authors employ SP to propose a new similarity search approach in the context of smart manufacturing. SP is also adopted to enhance automatic plausibility check [23, 24, 25]. In these papers, SP is adopted to project not just the component geometry, but also other relevant information, i.e. the settings and results of the simulations.

Based on this literature review, SP appears to be an appealing methodology. Translating point clouds into a uniform geometrical description, it provides a unique signature that can be representative of both simple and complex 3D geometries.

3 METHODOLOGY

In the early stage development of a vehicle, data and information coming from previous similar studies can be of aid. This idea lays out the basis for the following methodology, schematically represented in Fig. 2. The methodology is divided into two main parts: the mechanical problem and the ML one. In the former, the study case is set with respect to the original motivation. Here, the FE simulations are performed, and the results collected. These results form the datasets to be used in the latter part of the methodology where three different domain adaptation approaches are implemented.

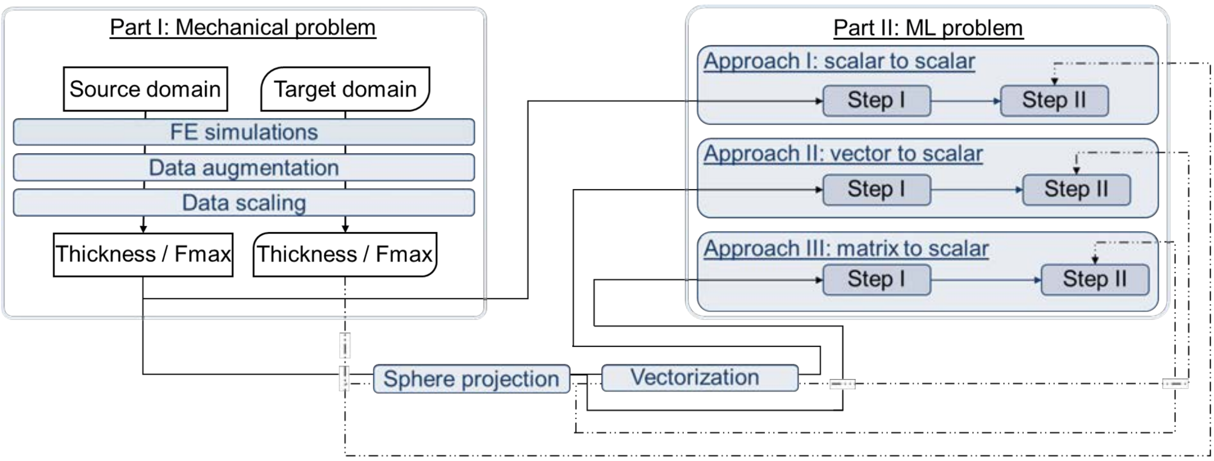

Figure 2: Scheme of the proposed methodology.

3.1 Mechanical problem

In the first part of the methodology, we set up the mechanical problem. Following the original motivation, we propose an uncertainties propagation study in a vehicle safety application of industrial relevance. The mechanical system that we decide to focus on is the vehicle bonnet. In this paper, we refer to the bonnet as the assembly constituted by different components, e.g. the upper and lower metal panels, the lifting mechanism, etc.

For the purpose of this study, the bonnet case is considered as an explanatory example. Considering the bonnet in a situation of full frontal impact, the goal is to gain insights into the relation between variations in panel thickness and the structural safety performance. Specifically, we focus on how the uncertain thickness value of the lower panel influences the intrusion of the bonnet into the window frame. Ideally, for occupant safety reasons, a crash should cause no such intrusion. This type of investigation is effectively tackled in industry during the architecture development of a new vehicle, e.g. in the body-in-white development process.

From a FE simulation perspective, it is hard to quantitatively measure the intrusion of the bonnet into the window frame. Generally, it is not only a matter of maximum displacement, but also involves the angle of intrusion. It is intuitive to picture that, according to the angle of impact, the bonnet can or cannot cause the breaking of the window frame. Given these circumstances, we simplify the mechanical problem. We do not regard the intrusion, but instead use the maximum force as quantity of interest.

Our mechanical problem is now to find the influence of the uncertain thickness of the lower panel on the maximum crushing force. For this purpose, we set up a parametrized FE simulation of the bonnet assembly. While the base of the lifting mechanism is fully constrained, an L-shaped rigid barrier moves with a fixed velocity of 56 km/h towards the bonnet in a full-frontal and symmetrical configuration. As per the problem statement, the thickness of the lower panel is considered as input variable that can be set by the user; the maximum force measured on the barrier, instead, represents the quantity of interest which is collected from the simulation results.

After setting up the simulations, the geometries representative of the source and target domain are selected; then, the FE simulations are performed using the R9.3.0 version of the LS-DYNA solver. The two different variants of car bonnet differ in terms of the topology in the lower panel; the other components constituting the assembly remain the same in terms of material, dimensions, and geometry. To one of the variants is assigned the role of source domain or, in this study case, of predecessors, from which the TL is supposed to learn the thickness–force relationship; the other variant is assumed to be the target domain, which is the new product under development to which the knowledge has to be transferred. The assumption underlying the study is that more data are available for the source domain, and fewer for the target domain.

To recall the uncertainties problem, FE simulations are run varying the value of the lower panel thickness in a range of $\pm 20\%$ from the nominal value. For both source and target domain bonnets, we run a total of 100 FE simulations. The Latin Hypercube technique is used to sample the design variable.

After performing the FE simulations, the collected datasets — thickness–force relationships for source and target domain — are augmented. Specifically, we compute two polynomial predictors and use them to generate further data points. The quality of the computed predictors is high enough (coefficient of determination greater than 0.99) to consider the newly augmented data points together with the ones generated through the FE simulations. Data augmentation is performed to make the datasets extensive enough for the following ML procedure, avoiding to pay a higher price in FE computational resources.

We now have 600 data points for each domain. Before moving to the second part of the methodology, the input/output data are scaled to zero mean and unit variance. While the range of the input is the same for both the domains, the output values of the target domain are scaled using the same mean and variance values computed for the source domain. This procedure is intended to improve the quality of the upcoming ML procedure.

3.2 Machine learning problem

At the end of the first part of the methodology, the data are collected and stored. The aim of the second part is to find the data shape that is most suited for the TL method and the application at hand. According to the basic assumption of TL, only few pieces of information are considered to be at disposal for the target system of interest. Therefore, we assume to use the source domain dataset entirely, while we consider available only a subset of the target domain data points. To draw the limited set of training points from the pool of previously generated target domain data, we use the stratified sampling technique [26].

In the following sections, we investigate three different approaches that are respectively referred to as scalar-to-scalar, vector-to-scalar and matrix-to-scalar approach. They share the underlying idea, but differ in terms of the input data shape. The concept of TL is translated, from a practical perspective, in a two steps procedure. First, a neural network (NN) is trained based on the source domain data; second, the same NN is adapted to the limited dataset of the target domain. The methodology proposed for the three approaches is similar to the one described in [10]. The implementation is done in Python 3.9 and the details of the approaches, as well as the assessment methods employed to evaluate their performance, are singularly addressed in the following.

The quality of the results of this study is evaluated in terms of the coefficient of determination R^2, which is computed for both TL and assessment methods. Considering a sample of n data points, $Y = [y_1, y_2, ..., y_n]$ is the vector of observed values, $\overline{y}$ is its mean value and $\hat{Y} = [\hat{y}_1, \hat{y}_2, ..., \hat{y}_n]$ is the vector of the predicted values. The R^2 is defined as:

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \overline{y})^2}. \tag{1}$$

The results of the three approaches are proposed for different sizes of the target domain datasets, respectively, $N_t = [3, 4, 5, 6, 7, 8, 9, 10, 15, 20]$ data points, with reference to the assumption of low data availability for the domain of interest. For testing, $N_v = 10$ systematically sampled target data points are used to assess the network performance. In each case, the R^2 coefficients are obtained for 20 independent, random training sets.

3.3 Approach I to the ML problem: scalar-to-scalar

Approach I is implemented using the high-level deep learning API Keras library [27]. The ML model of Approach I consists of a multilayer perceptron neural network (NN) with both single scalar input and output. With reference to the enounced mechanical problem of Section 3.1, the input is the scaled thickness value of the lower panel, while the output is the maximum force measured on the rigid barrier crashing on the bonnet.

In the first step of the procedure, the NN is trained on the source domain data. To this aim, we perform a gridsearch process [28], which results in an optimized network in terms of number of hidden layer, number of neurons per layer and learning rate. The NN is finally composed of 2 hidden fully connected layers with 10 neurons each. These settings, with rectified linear unit (relu) activation function and 0.03 learning rate, end up in a model able to make predictions on the source domain. After the first step, the network is ready to be adapted to the target domain.

In Approach I, the second step of the procedure — namely, the domain adaptation — follows two different modes that are here referred to as A and B. Domain adaptation in mode A is performed freezing the weights of the first NN layers and updating those of the last hidden layer. In mode B, with a procedure similar to BFTL-2 of [14], a further layer of 5 neurons is added at the end of the output layer of the initial NN, establishing a one–one mapping relationship between the output of the source NN, and the wanted quantity of interest for the target data.

The domain adaptation technique consists of 200 training epochs, in which either — mode A — the weights of the last layer change according to the new limited training target dataset, or — mode B — the last new layer is initialized and trained. The results, in terms of quality of the prediction, obtained at the end of the training process are compared to an assessment method specifically designed for this approach.

The assessment method that allows a fair comparison is a Gaussian process regression (GPR) model exclusively trained on the target domain training set. This choice follows the approach of [14] and represents the assumption of low data availability. With no exploitation of the data coming from previous designs — namely, the source domain — this method shows the prediction that would be obtained when only the target domain data are assumed to be available.

Approach I, with its simple architecture implementation, results in a straightforward methodology to treat the low complexity data of the problem at hand. However, it is common knowledge in the literature of TL that the outcome depends on the definition of the two domains [3]. For this reason, we decide to modify the input data shape, including the geometrical information of the source and target domains, leading towards the Approach II.

3.4 Approach II to the ML problem: vector-to-scalar

The source and target domain of this study are two bonnets that differ exclusively in terms of the topology of the lower panels. Approach II includes the description of the geometrical differences between the domains, resulting in an increased level of complexity for the NN input data shape with respect to Approach I.

The current problem, i.e. representing the geometrical information of the domains through ML readable data, sets new simple yet binding requirements. We demand a method that has to be applicable to complex components, giving flexibility to the whole TL framework. The eventually selected method has to manage mesh files in the input, and to provide a ML readable data in the output. Moreover, we wish the computational effort of the process to remain low. This makes the geometrical features representation used in [18], as well as other geometry classification methodologies, not viable. The method that appears to be the most suitable to quantitatively describe the geometrical differences between source and target domain is the sphere projection — SP — technique.

The SP is applied to the meshes of the two components that differ between the domains. Performing the projection procedure, explained in Section 2.2, information regarding the way the nodes are connected, i.e. elements of the FE mesh, is lost. Moreover, being the lower panels of the bonnets meshed with shell elements, SP results in 36x36 2D sparse matrices. In order to avoid zeros entering the NN, this problem is tackled by considering the 12x36 central rows of the matrices, that is later reshaped in the form of a 432x1 vector.

Treating the 2D matrix obtained from the SP as a 1D vector, we replicate the approach of [21]. Approach II is implemented using the Keras library [27], and relies on a multilayer perceptron NN with a vector of scalars in input and a scalar output. The input vector is composed of 433 elements, 432 of which are obtained through SP. To these, a further scalar element representing the panel thickness is appended. Approach II, similarly to Approach I, is composed of two steps: in the first, the NN is trained on the source domain data; in the second, it is adapted to the target domain.

The result of the first step is a NN with 5 fully connected hidden layers, 17 neurons per layer and 0.002 learning rate. In the second step, domain adaptation is implemented as in mode

A of Approach I. The last hidden layer is retrained through 200 new epochs and its weights are consequently modified. The assessment method of Approach II consists in a multilayer perceptron NN optimized and trained from scratch on the target domain training set.

In Approach II, SP is exclusively used to obtain a unique representation of the differences between source and target domain. The part of the input vector that comes from SP helps the NN to distinguish the domain of origin of the data points. Reshaping the 2D matrices obtained from the geometry projection in the form of a 1D vector allows to rely again on a multilayer perceptron NN, without moving to more complex-to-handle convolutional architectures. However, the spatial relationship between the elements of the matrix is lost. To overcome this limitation, and to consequently be more representative of the initial 3D geometries, we move to Approach III.

3.5 Approach III to the ML problem: matrix-to-scalar

Approach III is implemented in Pytorch [29]. Here, the 2D matrices obtained from the SP are kept in their original shape. Moving towards the implementation of a convolutional neural network architecture (CNN), the input shape of Approach III is given by two stacked 36x36 matrices, and the output remains the maximum force scalar value. While the first of the two input matrices comes from the projection of the geometry, the other is instead representative of the thickness distribution along the component nodes.

Similar to the approach proposed in [23], where SP is used to project not only geometry, but also the settings and the results of the FE simulations, we provide the thickness information in the form of a matrix by means of SP. On the pixels of the sphere, the number of projected nodes is multiplied by the respective value of thickness. As in Approaches I and II, also Approach III is constituted by a first step in which the CNN architecture is set and trained on the source domain, and a second step that gives space to the weights adaptation to the target domain data.

The CNN that is obtained at the end of the first step is composed of two convolutional layers — Conv1 and Conv2 — and two maximum pooling layers used for the feature extraction from the 2D matrices. The convolution layers are followed by two fully connected layers — Fc1 and Fc2 — and an output one — Out. After training on the source domain data with relu activation function, the CNN is ready for the domain adaptation procedure.

The domain adaptation procedure of Approach III follows the scheme of Table 1. Three combinations of frozen/unfrozen — F/U — layers are tested in each mode — Mode 0 to 2. 200 training epochs are performed to let the unfrozen layers of the CNN readapt to the limited training target dataset.

Domain adaptation mode	Conv1	Conv2	Fc1	Fc2	Out
Mode 0	F	F	U	U	U
Mode 1	F	U	U	U	U
Mode 2	F	U	U	U	F

Table 1: Modes of domain adaptation of Approach III.

The assessment method that is used to evaluate the outcome of Approach III is a CNN trained from scratch on exclusively the target domain data. The architecture of the assessment CNN is chosen to be the same as the one of the CNN trained on the source domain. As in Approach I and II, the assessment method is again representative of the prediction that would be obtained if only the target domain data were available.

4 RESULTS

Results are shown in Fig. 3. For Approach I, mode A and B are reported; for Approach III, modes 0, 1 and 2. Although referred to different methodologies — GPR for Approach I, NN for Approach II and CNN for Approach III — the assessment method is always depicted as TD in the results.

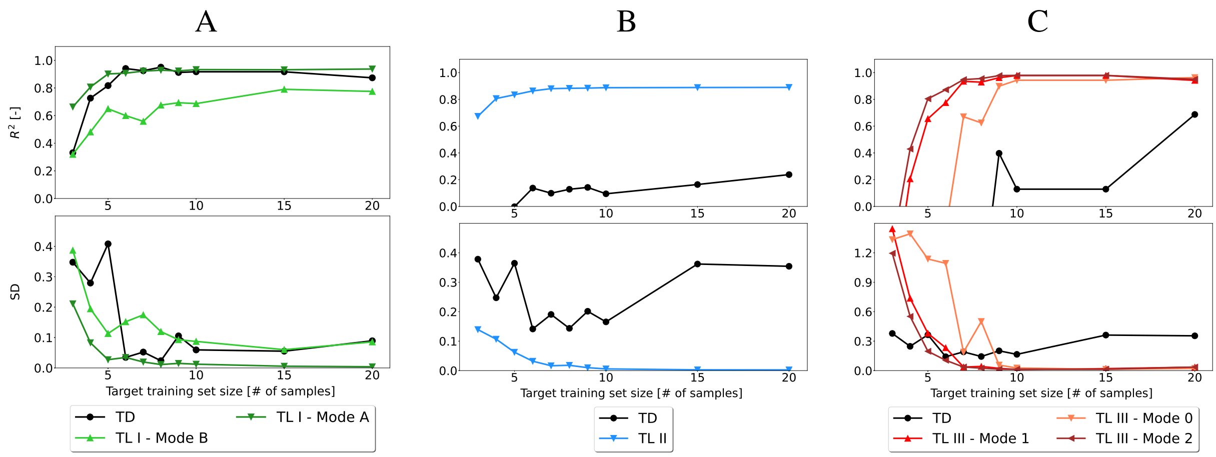

Figure 3: Results for Approach I (A), II (B) and III (C).

In Fig. 3, both the mean R^2 value as well as its standard deviation SD, out of the 20 repetitions, are provided for the three approaches. While the mean shows the quality of the performance in the approach, the SD provides information about how dispersed the data is with respect to the mean. In this paper, we regard the SD value as an indication of the repeatability of the proposed methodologies.

Overall, one can observe that, the fewer target domain data points are available, the more the quality of the prediction decreases. This fits with the general expectation according to which, with less information from the target domain, it becomes more difficult to compute a good prediction. Overall, however, TL achieves either better repeatability — Approach I — or higher R^2 values – Approaches II and III — with respect to the assessment methods.

Mode 2 of Approach III is the method that, out of all the tested ones, brings the best outcome. Approach III is in general outperforming Approach I that, in turn, achieves better results than Approach II. This suggests that the reframing of the 2D geometry matrix performed in Approach II is ineffective to the aim of TL. As confirmed by the results of Approach III, treating the 2D projected geometry in its original matrix shape seems to be the best choice.

Fig. 4A underlines the difference between mode A and B of Approach I. It suggests that adding a further layer — mode B — is less efficient than modifying the existing one — mode A. This conclusion is supported by the higher R^2 value obtained in mode A with respect to mode B for every size of the target training dataset. For training sets bigger than 6 data points, the difference in terms of R^2 value between TL mode A and the assessment method is almost unnoticeable. However, the SD shows that TL results are less dispersed than the ones of the assessment method, ensuring a higher repeatability of the methodology.

Fig. 4C allows the comparison between mode 0, 1 and 2 of Approach III. Looking at the values of the coefficient of determination, mode 2 is the best one, while mode 0 appears to be the worst. Mode 2 reaches an $R^2 = 0.81$ with only 5 data points. Mode 0 needs almost double the points to achieve the same performance. For a dataset with more than 7 data points, mode

1 and 2 show similar outcomes both in terms of R^2 and SD. We can conclude that keeping frozen the first convolutional and the last output layer ends up to be the best option for domain adaptation procedure.

The results of this paper are obtained for extremely limited target datasets — less than 20 points. TL is expected to provide significant advantages with respect to the assessment methods especially in these situations. With mode 2 of Approach III, where the maximum quality is indeed reached, only 9 target points suffice to achieve an R^2 value of 0.98.

5 CONCLUSION

This paper presents the first step towards the enhancement of a TL framework in crash analysis by means of the SP technique. The results of this paper have shown the attractive potential of SP-enhanced TL to transfer the knowledge from a source domain to a target domain. The quality of the obtained results is superior to the one of the assessment methods, especially for Approaches II and III which involve SP. This positive outcome supports the idea of using TL merged with SP in industrial scenarios that are characterized by low data availability.

In the automotive industry, when assessing the crash safety early on in the development, the final characteristics of a product are not yet entirely defined, resulting in unfeasible UQ studies. In this type of situations, when only limited amount of information is available on the product of interest, TL has been demonstrated to be of aid. By exploiting the high availability of the source domain data or, in our study, the predecessor designs, TL has been employed to fill the knowledge gap in the target domain. Based on our results, SP has helped this transfer of knowledge.

With respect to literature TL studies, we have demonstrated the possibility to acquire knowledge on a specific geometrical space, i.e the source domain bonnet, and to transfer it to another one. In comparison to the first attempt in this direction in [18], the method proposed in this paper allows more flexibility in the geometry description. This opens up the possibility to consider multiple topologies constituting the source domain, instead of a single one. Nonetheless, our method is expected to show its entire potential when applied to more complex and multi-inputs study cases. Moreover, this work also offers other directions of improvement, e.g. extending the inputs to include the projection of the FE boundary conditions, loads or connections. This is expected to allow the application of TL to cases where source and target domain differ of more than one component.

In conclusion, TL can serve as a valid alternative to expert knowledge for multipurpose metamodeling. We have shown that it is crucial to give an appropriate description of the domains and make an accurate selection of the ML approach. Enabling not just uncertainty quantification, but also V&V, robustness analysis, or feasibility studies, TL is confirmed as an appealing method to be used in crashworthiness studies. The final outlook expected from this research is to provide engineers with another useful tool to support their expertise and automatize the early-stage development operations.

REFERENCES

[1] H. Liu, S. Xu, and X. Wang, “Sampling strategies and metamodeling techniques for engineering design: Comparison and application,” *Proceedings of ASME Turbo Expo*, Seoul, South Korea, June 13-17, 2016.

[2] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, pp. 1–34, 2020.

[3] E. Otovic, M. Njirjak, D. Jozinovic, G. Mausa, A. Michelini, and I. Stajduhar, "Intra-domain and cross-domain transfer learning for time series data – how transferable are the features?," *Knowledge-Based Systems*, vol. 239, 2022.

[4] L. Torrey and J. Shavlik, "Transfer learning," *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*, pp. 242–264, 2010.

[5] H. Tercan, A. Guajardo, J. Heinisch, T. Thiele, C. Hopmann, and T. Meisen, "Transfer-learning: Bridging the gap between real and simulation data for machine learning in injection molding," *Procedia CIRP*, vol. 72, pp. 185–190, 2018.

[6] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," *Neurocomputing*, vol. 312, pp. 135–153, 2018.

[7] R. Liu, Y. Shi, C. Ji, and M. Jia, "A survey of sentiment analysis based on transfer learning," *IEEE Access*, pp. 85401–85412, 2019.

[8] L. Shao, F. Zhu, and X. Li, "Transfer learning for visual categorization: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 5, pp. 1019–1034, 2015.

[9] D. Sarkar, R. Bali, and T. Ghosh, *Hands-On Transfer Learning with Python*. Packt Publishing, 2018.

[10] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition*. O'Reilly Media, Inc., 2019.

[11] P. Cao, S. Zhang, and J. Tang, "Preprocessing-free gear fault diagnosis using small datasets with deep convolutional neural network-based transfer learning," *IEEE Access*, vol. 6, pp. 26241–26253, 2018.

[12] M. K. Ferguson, A. Ronay, Y.-T. T. Lee, and K. H. Law, "Detection and segmentation of manufacturing defects with convolutional neural networks and transfer learning," *Smart and sustainable manufacturing systems*, vol. 2, 2018.

[13] J. Jung, Y. Kim, J. Park, and S. Ryu, "Transfer learning for enhancing the homogenization-theory-based prediction of elasto-plastic response of particle/fiber-reinforced composites," *Composite Structures*, vol. 285, 2022.

[14] Subhayan, J. Britton, M. Reynolds, R. Skinner, K. Jansen, and A. Doostan, "On transfer learning of neural networks using bi-fidelity data for uncertainty propagation," *International Journal for Uncertainty Quantification*, pp. 1–29, 2019.

[15] D. Maturana and S. Scherer, "Voxnet: A 3d convolutional neural network for real-time object recognition," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 922–928, Hamburg, Germany, Semptember 28 - October 2, 2015.

[16] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3d shape recognition," *Proceedings of the IEEE international conference on computer vision*, Santiago, Chile, December 7-13, 2015.

[17] X. Guo, W. Li, and F. Iorio, "Convolutional neural networks for steady flow approximation," *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 481–490, San Francisco, California, August 13-17, 2016.

[18] G. Colella, V. A. Lange, and F. Duddeck, "Transfer learning for metamodel construction to enable uncertainty quantifications in crash design based on scarce data availability," *ISMA-USD Proceedings*, Leuven, Belgium, September, 2021.

[19] T. C. Spruegel and S. Wartzack, "Konzept zur automatischen Bauteilerkennung von Normteilen in ANSYS mechanical mittels künstlichen neuronalen Netzen," *ANSYS Conference & 32. CADFEM Users Meeting*, Nuremberg, Germany, June 4, 2014.

[20] T. C. Spruegel and S. Wartzack, "Konzept zur automatischen Bauteilerkennung innerhalb der FE-Software Umgebung mittels künstlichen neuronalen Netzen," *DfX-Symposium, Hamburg:TuTech*, Bamberg, Germany, October, 2014.

[21] T. C. Spruegel and S. Wartzack, "Concept and application of automatic part-recognition with artificial neural networks for fe simulations," *International conference on engineering and design (ICED)*, Milano, Italy, July 27-30, 2015.

[22] S. Bickel, C. Sauer, B. Schleich, and S. Wartzack, "Comparing CAD part models for geometrical similarity: A concept using machine learning algorithms," *Procedia CIRP*, vol. 96, pp. 133–138, 2021.

[23] T. Spruegel, T. Schroeppel, and S. Wartzack, "Generic approach to plausibility checks for structural mechanics with deep learning," *21st International conference of engineering and design (ICED)*, Vancouver, Canada, August 21-25, 2017.

[24] T. C. Spruegel, R. Rothfelder, S. Bickel, A. Grauf, C. Sauer, and B. Schleich, "Methodology for plausibility checking of structural mechanics simulations using deep learning on existing simulation data," *Proceedings of NordDesign*, Linkoeping, Sweden, August 14-17, 2018.

[25] T. C. Spruegel, S. Bickel, B. Schleich, and S. Wartzack, "Approach and application to transfer heterogeneous simulation data from finite element analysis to neural networks," *Journal of Computational Design and Engineering*, vol. 8, no. 1, pp. 298–315, 2021.

[26] A. L. Russell, "The design of social research," *The University of Chicago press Journals*, vol. 28, no. 3, 1953.

[27] F. Chollet *et al.*, *Keras*. GitHub, 2015.

[28] F. Pedregosa, V. Gaël, G. Alexandre, M. Vincent, and et al., "Scikit-learn: Machine learning in python," *The Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[29] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library.," *Advances in Neural Information Processing Systems*, vol. 32, pp. 8024–8035, 2019.